Stochastic Dynamic Bin Packing in a Large-Scale Service System

Yuan Zhong
Columbia University

Joint works with Sasha Stolyar, Lehigh University
Outline

• Motivation

• Model Description

• Greedy-with-Sublinear safety Stock (GSS) algorithms
  – Why naïve Greedy may not work well
  – Asymptotic optimality

• GRAND algorithms
  – Asymptotic optimality of GRAND(aZ)
  – Simulations

• Discussion
Virtual Machine Placement in a Cloud

Virtual machines

Physical machines
Packing Constraints

VM

A typical PM

Storage

CPU
Packing Constraints

VM

A typical PM

Storage

CPU
Packing Constraints

VM

A typical PM
Feasible Configuration

- Infinite number of homogeneous servers.
- $l$ types of VMs (ex. $l = 2$).
- Feasible configurations $k = (k_1, k_2, ..., k_i, ..., k_l)$ on a server.
- $k = (1, 1)$ in example.
Feasible Configuration

- Infinite number of homogeneous servers.
- \( I \) types of VMs (ex. \( I = 2 \)).
- Feasible configurations \( k = (k_1, k_2, ..., k_i, ..., k_I) \) on a server.
  - \( k = (1, 1) \) in example.
  - \( k = (1, 2) \) NOT feasible!
Feasible Configuration

- Infinite number of homogeneous servers.
- $I$ types of VMs (ex. $I = 2$).
- Feasible configurations $k = (k_1, k_2, \ldots, k_i, \ldots, k_I)$ on a server.
- $k = (1, 1)$ in example.
- Ex. of vector packing.
- $X_k$ : no. of servers with configuration $k$. 

Type 1 Type 2

Storage CPU

No. of type-i VMs
Infinite Server Model

- An abstract set $K$ of feasible configurations:
  - Monotonicity: if $k \in K$ and $k' \leq k$, then $k' \in K$.
  - More general than vector packing constraints.
Infinite Server Model

• An abstract set $K$ of feasible configurations:
  – Monotonicity: if $k \in K$ and $k' \leq k$, then $k' \in K$.
  – More general than vector packing constraints.

• Key assumption: service times independent across customers!
  – E.g. ignores communication between customers/VMs.

Minimize no. of occupied servers subject to packing constraints
• 2 types of VM; \( I = 2 \).

• \( \rightarrow \) type-1 arrival.

• Before arrival:
  • \( X_{(2,2)} = 3, X_{(3,2)} = 4 \).

• Dots are feasible conf.

• NOT vector packing constraints.
• 2 types of VM; \( I = 2 \).

• Type-1 arrival.

• \( X_{(2,2)} := X_{(2,2)} - 1 = 2 \);

• \( X_{(3,2)} := X_{(3,2)} + 1 = 5 \).

• Change in no. of occupied servers occur only at \( e_i \).

• Can’t place arrival into conf. \( k \) with \( X_k = 0 \), \( k \neq 0 \).
Asymptotic Approach

- Exact problem NP-complete; look for asymptotic optimality.

- Define *fluid-scale* processes: \( x_k^{(r)}(t) = X_k^{(r)}(t)/r \).

- \( x_k^{(r)}(t) \longrightarrow x_k(t) \) u.o.c as \( r \longrightarrow \infty \). Optimal set \( X^* \) solves:

\[
\text{Minimize } \sum_k x_k \\
\text{subject to } \sum_i k_i \mu_i x_k = \lambda_i \text{ for all } i; \\
\quad x_k \geq 0 \text{ for all } k.
\]

Population Conservation:
arrival rate ≈ departure rate in steady state
Brief Literature Review

Systems:
- [Gulati et al ’12]
- [Ambrust et al ’09]
- References therein

Parallel Server Queueing Systems:
- No packing constraints
- Min queueing delay
- Huge literature

This work:
- Infinite # of bins/servers
- Customer departures
- General packing constraints

Classical Online Bin Packing:
- No departures
- Specific packing constraints
- [Bansal et al ’09], [Csirik et al ’06], etc

Packing Service Systems:
- Finite # of bins
- Max throughput/min queueing delay
- [Jiang et al ’12], [Maguluri et al ’12], etc

Dynamic Bin Packing:
- [Coffman et al ’83]
- 1-dimensional
- Non-stochastic
Some Simple Packing Policies

• Naïve greedy packing:
  • Place into any fitting server.

• Random greedy packing:
  • Place into a fitting server, unif. at rand.

• Best-fit packing:
  • One resource: place into most loaded fitting server.
  • Multiple resources: not clear what to do.
Problem with Naïve Greedy

- Consider a closed system
- $X_{(3,3)} = 70$; other $X_k = 0$.
- **type-1 arrival.**
- **type-1 departure.**
- Departure from (3, 3)
  - Immediately replaced an type-1 arrival
Problem with Naïve Greedy

- type-1 arrival.
- type-1 departure.
- \( X_{(3,3)} := X_{(3,3)} - 1 = 69. \)
- \( X_{(2,3)} := X_{(2,3)} + 1 = 1. \)
Problem with Naïve Greedy

- type-1 arrival.
- type-1 departure.
- Departure replaced immediately.
- $X_{(3,3)} := X_{(3,3)} - 1 = 70$.
- $X_{(2,3)} := X_{(2,3)} + 1 = 0$.
- Stuck in this state.
Problem with Naïve Greedy

- Better state:
  - $X_{(6,1)} = X_{(1,6)} = 30$;
  - $X_k = 0$ otherwise.
  - $\sum_k X_k = 60 < 70$. 

**Diagram**: 
- The diagram shows a grid with points marked at $(1,6)$, $(3,3)$, and $(6,1)$. 
- The line $k_2 = -(1/5)k_1 + 70$ is shown, indicating the boundary for the optimal solution. 
- The points $(1,6)$ and $(6,1)$ are on the line, suggesting they are part of the optimal solution.
Problem with Naïve Greedy

- Better state:
  - $X_{(6,1)} = X_{(1,6)} = 30$;
  - $X_k = 0$ otherwise.
  - $\bigwedge_k X_k = 60 < 70$.

- Can have bad conf. surrounded by 0s.

- May want to create “safety stocks” automatically for better “conductance”.
  - Move from bad state to good state.
Greedy Algorithm in [Stolyar ’12]

- Naïve Greedy really tries to greedily minimize $\sum_k x_k$
- [Stolyar ’12] greedily drive down approximate objective $\sum_k x_k^{1+\alpha} \left( \frac{\alpha}{\alpha \rightarrow 0} \sum_k X_k \right)$

- Arrival $k \rightarrow k+e_i$.
- $X_{k+e_i} := X_{k+e_i} + 1; X_k := X_k - 1$.
- $\Delta$ in obj. $\approx X_{k+e_i}^{1+\alpha} - X_k^{1+\alpha}$.
- Place arrival so $\Delta$ is min.
Greedy Algorithm in [Stolyar ’12]

- Convexify $\sum_k x_k$ to $\sum_k x_k^{1+\alpha}$, and greedily decrease $\sum_k x_k^{1+\alpha}$ instead.

- GREEDY: place an i-arrival into a server with conf $k - e_i$ so that:
  - Feasibility: $k - e_i$ is feasible;
  - Availability: either $x_{k-e_i} > 0$ or $k-e_i = 0$;
  - Smallest increment: $x_k^\alpha - x_{k-e_i}^\alpha$ is smallest.

- Can show $\frac{d}{dt} (\sum_k x_k^{1+\alpha}) \leq 0$; and $< 0$ whenever $x$ is away from CVX opt.

- $O(r)$ safety stocks everywhere.
  - $O(r)$ loss from optimum.
  - Want sublinear safety stocks!
Greedy with sublinear Safety Stocks (GSS)

- Create $O(r^p)$ safety stocks.
- Define local-fluid-scale states:
  - $\hat{x}_k^{(r)} = X_k^{(r)}/r^p \quad (p \in (\frac{1}{2}, 1))$.
- $w(\hat{x}) = \min\{1, \hat{x}\}$.
- $\otimes(k, i) = w(\hat{x}_k^{(r)}) - w(\hat{x}_{k-e_i}^{(r)})$.
- Place type-i arrival into $k-e_i$ (feasibility) with minimal $\otimes(k, i)$ (min increment) and [either $\hat{x}_{k-e_i}^{(r)} > 0$ or $k-e_i = 0$] (availability).
Greedy with sublinear Safety Stocks (GSS)

- Greedily minimize
  \[ F(\hat{x}) = \bigotimes_k f(\hat{x}_k), \text{ where} \]
  \[ f(\hat{x}) = \begin{cases} 
  \frac{\hat{x}^2}{2}, & \hat{x} \leq 1; \\
  \frac{1}{2}, & \hat{x} > 1.
\end{cases} \]

- \( df(\hat{x})/d\hat{x} = w(\hat{x}) \)

- \( \bigotimes F(\hat{x}) \) captured through \( \{ \bigotimes(k, i) \} \).

- \( F \approx \bigotimes_k x_k \).
Greedy with sublinear Safety Stocks (GSS)

\[ \mathbb{1}(k, 1) \]

\[ k - e_1 \rightarrow k \]

Plot of \( f(\hat{x}) \) against \( \hat{x} \)

\[ \frac{1}{2} \text{ (local fluid)} \]

\[ (= r^{p-1} \rightarrow 0 \text{ on fluid scale}) \]
Theorem (S and Z.)

Consider closed infinite-server systems with monotone packing constraints, indexed by \( r \). For each \( r \), let \( x_k^{(r)}(\square) \) be the stationary probability distribution of the \( r^{\text{th}} \) system. Then under GSS,

\[
x_k^{(r)}(\square) \xrightarrow{\text{distribution}} X^* \text{ in distribution, as } r \rightarrow \infty.
\]

(Recall \( X^* \) is the set of optimal solutions to LP)

- Recall LP: Minimize \( \sum_k x_k \) subject to \( \sum_k k_i x_k = \lambda_i / \mu_i \) for all \( i \); \( x_k \geq 0 \) for all \( k \).

- Proof follows from the next two lemmas.
Greedy Algorithm of [Stolyar ’12]

• [Stolyar ’12] studied the same model
• Approach: greedily drive down some $\mathbb{C}_k f(x_k) (\approx \mathbb{C}_k X_k)$
  – Naïve greedy $\rightarrow \mathbb{C}_k X_k$
    • can get stuck at subopt. solution.
  
  – [S ’12]: $\mathbb{C}_k X_k^{1+\alpha} \xrightarrow[\alpha \rightarrow 0]{\alpha} \mathbb{C}_k X_k$

  – [S and Z ’13]: $\mathbb{C}_k X_k + O(r^p) \approx \mathbb{C}_k X_k$ at fluid scale.

  – Asymptotic opt. at fluid scale.
Some Simple Packing Policies

- Naïve greedy packing:
  - Place into any fitting server.

- Random greedy packing:
  - Place into a fitting server, unif. at rand.

- Best-fit packing:
  - One resource: place into most loaded fitting server.
  - Multiple resources: not clear what to do.

- Main takeaway: Random packing works well!
  - Departures undo “bad” decisions.
Greedy Random (GRAND) Policies

General GRAND

At each time $t$:

- Designate $X_0(t) = f(X(t))$ empty servers — zero servers.
- Type-$i$ incoming: $X^{(i)}(t) = X_0(t) + \# \text{ occupied fitting servers}$
- If $X^{(i)}(t) > 0$
  - Place into one of these servers, unif. at rand$^1$.
- Else
  - Place into an empty server.

- **GRAND**$(aZ)$: $X_0 = aZ$ ($Z$ = total no. of customers)
  - Implementation is easy — only keep track of $Z$.
  - Not obvious: GRAND$(aZ)$ ($a > 0$) drives down an obj. func.
Obj. Func. of GRAND(aZ) \( (a > 0) \)

- Define \( c_k = (k_1!)(k_2!)...(k_I!) \).

- \( L^{(a)}(x) = b^{-1} \sum_k x_k \log\left[ x_k c_k / (ea) \right] \) where \( b = -\log a \).
  - \( L^{(a)}(x) \to \sum_k x_k \) as \( a \to 0 \).
  - \( L^{(a)}(x) \) entropy-like: GRAND(aZ) “mixes” states so \( L^{(a)} \) decreases.

- \( x^{*,a} \) is the unique opt. of

\[
\begin{align*}
\text{Minimize} & \quad L^{(a)}(x) \\
\text{subject to} & \quad \sum_k k_i \mu_i x_k = \lambda_i \quad \text{for all } i; \\
& \quad x_k \geq 0 \quad \text{for all } k.
\end{align*}
\]
Main Results

Theorem 1 (S and Z.)
Let $a > 0$. Consider a sequence of systems under GRAND($aZ$), indexed by $r$. For each $r$, let $x^{(r)}(\square)$ be the stationary probability distribution of the $r^{th}$ system. Then,

$$x^{(r)}(\square) \xrightarrow{\text{in distribution}} x^{*,a} \text{ as } r \to \infty.$$

Theorem 2 (S and Z.)
As $a \to 0$, $x^{*,a} \to X^*$.

- Recall $X^*$ is the set of opt. of LP – characterizes min fluid-scale
  # of occupied servers
- Theorem 2 follows from $L^{(a)}(x) \xrightarrow{\text{as } a \to 0} \bigoplus_k x_k$. 

Theorem 1: Proof Sketch

- Flow conserv. in steady state
  - All dept. allocated “back”

- Dept. rate $k \rightarrow k-e_i : k_i \mu_i x_k$.

- Fraction $x_{k'-e_i} / x(i)$ allocated “back” $k'-e_i \rightarrow k'$.

- $D_1 = k_i x_k x_{k'-e_i}; D_2 = k'_i x_{k-e_i} x_k'$.

- Change to $L^{(a)}$ is $\xi_{k,k',i}$
  $$k_1 = b^{-1} \left[ \log(D_2) - \log(D_1) \right] (\mu_i / x(i)) D_1$$

Due to $k'-e_i \rightarrow k'$
Due to $k \rightarrow k-e_i$
Theorem 1: Proof Sketch

- $\xi_{k,k',i}$ was for the pair $(k \rightarrow k-e_i, k'-e_i \rightarrow k')$.

- Similarly, $\xi_{k,k,i}$ for the pair $(k' \rightarrow k'-e_i, k-e_i \rightarrow k)$.

- $\xi_{k,k',l} + \xi_{k',k,l}$ proportional to $[\log(D_2) - \log(D_1)](D_1-D_2) \leq 0$.

- Boundary cases are okay.

- Summing these terms gives $(d/dt)L^{(a)}(x(t)) \leq 0.$
• System scale = \( r \) ≈ total # of customers

• Initial state: \( X_{(3,3)} = r/6 \)

• Opt solution:
  • \( X_{(1,8)} = X_{(8,1)} = r/18 \).
  • # occ. servers = \( r/9 \).
Simulations

- Opt # of occ. servers $\approx 1111$
- GRAND(0) seems to be perform best
Numerical Results

- Opt # of fluid-scale occ. servers $\approx 1/9$
- larger $a \rightarrow$ faster conv. speed but worse performance
Discussion

• Infinite-server model motivated by VM placement in cloud.

• Greedy Random (GRAND) policies
  – Extremely simple, easy to implement
  – Asymptotic optimality (as $r \rightarrow \infty$ and $a \rightarrow 0$)
  – Contrary to intuition from classical bin packing

• Conjecture: GRAND(0) is asymptotic optimal.

• In practice, random does not do very well.
  – Servers are highly heterogeneous.
  – Meaningful GRAND extension to heterog. server populations?